AN O(n) ALGORITHM FOR QUADRATIC KNAPSACK PROBLEMS

Peter BRUCKER *

Fachbereich Mathematik, Universität Osnabrück, Postfach 4469, 4500 Osnabrück, Federal Republic Germany

Received August 1983
Revised May 1984

An algorithm is presented which solves bounded quadratic optimization problems with n variables and one linear constraint in at most O(n) steps. The algorithm is based on a parametric approach combined with well-known ideas for constructing efficient algorithms. It improves an O($n \log n$) algorithm which has been developed for a more restricted case of the problem.

nonlinear programming * convex programming * quadratic programming * knapsack problem * parametric programming

1. Introduction

Consider the following quadratic program

$$\text{P:}\quad \min \sum_{i=1}^{n} \left(\tfrac{1}{2}d_i x_i^2 - a_i x_i\right) \tag{1}$$

$$\text{s.t.}\quad \sum_{i=1}^{n} b_i x_i = b_0, \tag{2}$$

$$l_i \leqslant x_i \leqslant u_i,\; i = 1,\dots,n, \tag{3}$$

where $d_i > 0$ for $i = 1,\dots,n$.

In P we may also assume that $b_i > 0$ for $i = 1,\dots,n$ because (i) if $b_i = 0$ then x_i can be calculated independently from (2) and the corresponding term in (1) and (3) my be eliminated, and (ii) if $b_i < 0$ we may replace x_i by $-x_i$, b_i by $-b_i$, a_i by $-a_i$, l_i by $-u_i$, and u_i by $-l_i$ to get essentially the same problem.

Under the special assumption that $b_i = 1$ and $l_i = 0$ for all $i = 1,\dots,n$ Helgason et al. [2] derived an O($n \log n$) algorithm for solving P. We will show that the more general problem P can be solved in linear time.

2. A parametric approach

For each parameter value $t \in \mathbb{R}$ we consider the problem

$$\text{P}(t):\quad \min \sum_{i=1}^{r} \left(\tfrac{1}{2}d_i x_i^2 + (b_i t - a_i) x_i\right) \quad \text{s.t.}\quad l_i \leqslant x_i \leqslant u_i,\quad i = 1,\dots,n. \tag{4}$$

P(t) has the unique solution $x(t)$ with

$$x_i(t) = \begin{cases} l_i & \text{if } (a_i - b_i t)/d_i \leqslant l_i, \\ u_i & \text{if } (a_i - b_i t)/d_i \geqslant u_i, \\ (a_i - b_i t)/d_i & \text{otherwise.} \end{cases} \tag{5}$$

Let $z : \mathbb{R} \to \mathbb{R}$ be the function defined by

$$z(t) = \sum_{i=1}^{n} b_i x_i(t). \tag{6}$$

Define *critical parameters* $t_i^{\mathrm{U}} \leqslant t_i^{\mathrm{L}}$ by

$$t_i^{\mathrm{L}} = (a_i - l_i d_i)/b_i \quad\text{and}\quad t_i^{\mathrm{U}} = (a_i - u_i d_i)/b_i$$

for $i = 1,\dots,n$ and let

$$t_1 < t_2 < \cdots < t_r \tag{7}$$

be all distinct critical parameter values. Furthermore let $t_0 = -\infty$, $t_{r+1} = +\infty$. Then within each interval (t_i, t_{i+1}), $i = 0,\dots,r$, the structure of $x(t)$ does not change. The following theorem states some useful properties of the function z.

Theorem. *The function $z : \mathbb{R} \to \mathbb{R}$ defined by* (6) *has the following properties*:

(a) *z is monotone not increasing.*

(b) *If* $z(t)=b_0$ *then* $x(t)$ *is an optimal solution of* P.

(c) *If* $z(t)>b_0$ *(resp.* $z(t)<b_0$*) for all* $t\in\mathbb{R}$ *then* P *has no feasible solution.*

Proof. We write

$$f(x)=\sum_{i=1}^{n}\left(\tfrac{1}{2}d_i x_i^2-a_i x_i\right),\quad bx=\sum_{i=1}^{n}b_i x_i$$

and

$$f(x,t)=f(x)+tbx.$$

(a) Assume that $t\in\mathbb{R}$, $t^*=t+\Delta t$ where $\Delta t>0$, and

$$z(t)=bx(t)<bx(t^*)=z(t^*).$$

Then optimality of $x=x(t)$ ($x^*=x(t^*)$) for P(t) (P(t^*)) implies

$$f(x)+tbx\leqslant f(x^*)+tbx^*$$
$$\left(f(x^*)+(t+\Delta t)bx^*\leqslant f(x)+(t+\Delta t)bx\right).$$

Thus we get the contradiction

$$f(x^*)+(t+\Delta t)bx^*\leqslant f(x)+(t+\Delta t)bx$$
$$<f(x^*)+(t+\Delta t)bx^*.$$

(b) Let x' be an arbitrary feasible solution of P. Then x' is feasible for P(t) and optimality of $x(t)$ for P(t) implies

$$f(x(t))+bx(t)\leqslant f(x')+bx'$$

which implies $f(x(t))\leqslant f(x')$ because $bx(t)=bx'=b_0$.

(c) Assume that $z(t)=bx(t)>b_0$ for all $t\in\mathbb{R}$ and that P has a feasible solution x. Then $bx=b_0$. Furthermore, there exists a parameter value t^* such that $x(t^*)$ is optimal for all $P(t)$ will $t\geqslant t^*$ (see (5)). Thus for all $t\geqslant t^*$ we have

$$f(x(t^*))+tbx(t^*)\leqslant f(x)+tbx,$$
$$f(x(t^*))-f(x)\leqslant t\left(bx-bx(t^*)\right)$$
$$=t\left(b_0-bx(t^*)\right)$$

which for large $t\geqslant t^*$ leads to a contradiction because $b_0-bx(t^*)<0$.

If $z(t)<b_0$ for all $t\in\mathbb{R}$ the desired result follows similarly. □

If $z(t_1)\geqslant b_0$ and $z(t_r)\leqslant b_0$ then there exists an index j with $z(t_j)\geqslant b_0$ and $z(t_{j+1})\leqslant b_0$. Using this index j an optimal solution may be constructed as follows. Define the index sets

$$I_L=\left\{i \mid t_i^L\leqslant t_j\right\},$$
$$I_U=\left\{i \mid t_{j+1}\leqslant t_i^U\right\},$$
$$I_M=\left\{i \mid t_i^U<t_{j+1};\ t_j<t_i^L\right\}.$$

Then for each $t\in[t_j,t_{j+1}]$ the components of the solution $x(t)$ may be expressed by

$$x_i(t)=\begin{cases} l_i & \text{if } i\in I_L,\\ u_i & \text{if } i\in I_U,\\ (a_i-b_i t)/d_i & \text{if } i\in I_M.\end{cases}\tag{8}$$

We find the optimal solution $x(t_{opt})$ of P by substituting in (8) the parameter value

$$t_{opt}=\left(\sum_{i\in I_U}b_i u_i+\sum_{i\in I_L}b_i l_i+\sum_{i\in I_M}\left(\frac{b_i a_i}{d_i}\right)-b_0\right)\Big/\sum_{i\in I_M}\frac{b_i^2}{d_i}.\tag{9}$$

Notice that t_{opt} is a solution of equation $bx(t)=b_0$.

Due to the monotonicity of the function z, the index j can be found by a binary search on the index set of the critical values (7). This leads to an O($n\log n$)-algorithm. In the next section we will show that the search for index j can be done in O(n) steps which leads to a linear time algorithm for P.

3. An O(n) search algorithm

The main idea of the algorithm may be described as follows. Let us assume that $z(t_1)\geqslant b_0$ and $z(t_r)\leqslant b_0$. If we set $t_{min}=t_1$ and $t_{max}=t_r$ we are sure that the property

$$t_{opt}\in[t_{min},t_{max}]\tag{10}$$

holds. During the algorithm, keeping property (10) satisfied, the interval $[t_{min},t_{max}]$ is decreased step by step until it has the form $[t_j,t_{j+1}]$. This is accomplished by solving P(t) for different values $t=t_i$. Let $t_{min}<t_i<t_{max}$. If $z(t_i)>b_0$ then we may replace t_{min} by t_i. If $z(t_i)<b_0$ then t_{max} is replaced by t_i. If $z(t_i)=b_0$ we have $t_{opt}=t_i$.

The important point is that at the same time the relevant critical parameter sets

$$T^U=\left\{t_i^U \mid i=1,\ldots,n\right\}\quad\left(T^L=\left\{t_i^L \mid i=1,\ldots,n\right\}\right)$$

are reduced by eliminating the endpoints of intervals $[t_i^U, t_i^L]$. By solving at most two of the problems $P(t^L)$ and $P(t^U)$ each of the sets T^U, T^L reduces at least by $\lceil \frac{1}{4}|T^U| \rceil$ ($\lceil \frac{1}{4}|T^L| \rceil$) elements. Furthermore when eliminating t_i^U, t_i^L we know the structure of $x_i(t_{opt})$ (i.e. we know whether $x_i(t_{opt}) = u_i$ or $x_i(t_{opt}) = l_i$ or $x_i(t_{opt}) = (a_i - b_i t_{opt})/d_i$). Thus the complexity for solving problems $P(t)$ reduces by the same fraction. For the overall complexity of the algorithm we get the upper bound

$$cn + \tfrac{3}{4}cn + \left(\tfrac{3}{4}\right)^2 cn + \cdots = 4cn = O(n)$$

where c is some constant. We choose t^L, t^U in such a way that

$$t^L = \text{median}(T^L)$$

and

$$t^U = \text{median}\left(\left\{ t_i^U | t_i^L \in T^L;\ t_i^L \geq t^L \right\}\right)$$

where median(S) denotes the median of set S. median(S) can be calculated in at most $O(|S|)$ steps (see Aho, Hopcraft and Ullman [1]).

We solve $P(t^L)$ ($P(t^U)$) only if $t_{min} < t^L < t_{max}$ ($t_{min} < t^U < t_{max}$) and update the values t_{min} and t_{max}. A quarter of the sets T^U and T^L can be eliminated and the structure of $x_i(t)$ is known for the corresponding variables for each $t \in [t_{min}, t_{max}]$. To see this we have to consider three cases.

Case 1: $t^L \leq t_{min}$. Then $t_i^L \leq t_{min} \leq t_{opt}$ for at least $\lceil \frac{1}{2}|T^L| \rceil$ intervals $[t_i^U, b_i^L]$. For these i we can eliminate t_i^U (t_i^L) from T^U (T^L) and fix $x_i(t) = l_i$ for all $t \in [t_{min}, t_{max}]$.

Case 2: $t^L > t_{min}$ and $z(t^U) \leq b_0$ (i.e. $t^U \geq t_{max}$). Then $t_{opt} \leq t_{max} \leq t^U$ and by construction of t^U for at least $\lceil \frac{1}{4}|T^L| \rceil$ intervals $[t_i^U, t_i^L]$ we have $t_{opt} \leq t^U \leq t_i^U$. For these i we can eliminate t_i^U (t_i^L) from T^U (T^L) and fix $x_i(t) = u_i$ for all $t \in [t_{min}, t_{max}]$.

Case 3: $t^L > t_{min}$ and $z(t^U) > b_0$. Then $t^U < t_{opt}$. Furthermore $t_{opt} \leq t^L$ because otherwise $t_{opt} > t^L > t_{min}$ which is contradicting the fact that t_{min} must have been updated after solving $P(t^L)$. Now $t^U \leq t_{opt} \leq t^L$ implies that for at least $\lceil \frac{1}{4}|T^L| \rceil$ intervals $[t_i^U, t_i^L]$ we have

$$t_i^U \leq t_{min} \leq t_{opt} \leq t_{max} \leq t_i^L.$$

For these i we can eliminate t_i^U (t_i^L) from T^U (T^L) and fix

$$x_i(t) = (a_i - b_i t)/d_i \quad \text{for all } t \in [t_{min}, t_{max}].$$

Details of the procedure are described by the following algorithm. Before applying the algorithm all critical values t_i^U, t_i^L are calculated. We assume that immediately before using $z(t)$ problem $P(t)$ is solved. Furthermore I is the current index set of variables $x_i(t)$ which are not fixed. Note that

$$T^U = \left\{ t_i^U | i \in I \right\} \quad \text{and} \quad T^L = \left\{ t_i^L | i \in I \right\}$$

are the current sets of parameter values.

Algorithm

```
    BEGIN
 1. IF z(t_1) = b_0 OR z(t_r) = b_0 THEN STOP (Solution
    optimal);
 2. IF z(t_1) < b_0 OR z(t_r) > b_0 THEN STOP (No
    feasible solution exists);
 3. t_min := t_1; t_max := t_r;
 4. I := {1,...,n};
 5. WHILE I ≠ ∅
    DO BEGIN
 6.   t^L := median({t_i^L | i ∈ I});
 7.   t^U := median({t_i^U | i ∈ I; t_i^L ≥ t^L});
 8.   FOR t := t^L, t^U IF t_min < t < t_max THEN
      DO BEGIN
 9.     IF z(t) = b_0 THEN STOP (Solution optimal);
10.     IF z(t) > b_0 THEN t_min := max{t_min, t}
11.     ELSE t_max := min{t_max, t}
      END DO
12.   FOR ALL i ∈ I
      DO BEGIN
13.     IF t_i^L ≤ t_min THEN
14.       BEGIN I := I\{i}; x_i := l_i END;
15.     IF t_max ≤ t_i^U THEN
16.       BEGIN I := I\{i}; x_i := u_i END;
17.     IF t_i^U ≤ t_min ≤ t_max ≤ t_i^L THEN
18.       BEGIN I := I\{i}; x_i(t) := (a_i - b_i t)/d_i
          END
        END DO
      END DO
    END.
```

If the algorithm does not stop in step 1, 2 or 9 we must have $t_j = t_{min}$, $t_{j+1} = t_{max}$ and the optimal solution may be calculated using (8) and (9). Finally note that $z(t^L)$ and $z(t^U)$ can be calculated in $O(|I|)$ steps. This can be seen as follows.

Let I^u and I^ℓ and I^m be the set of all indices eliminated from I during steps 16 and 14 and 18 respectively. Then at each stage of the algorithm we have

$$z(t)=\sum_{i\in I} b_i x_i(t)+\sum_{i\in I^u} b_i u_i+\sum_{i\in I^\ell} b_i l_i + \sum_{i\in I^m} a_i b_i/d_i-\sum_{i\in I^m}\left(b_i^2/d_i\right)t$$

for all $t\in[t_{\min}, t_{\max}]$.

Thus, if the sums

$$\sum_{i\in I^u} b_i u_i,\quad \sum_{i\in I^\ell} b_i l_i,\quad \sum_{i\in I^m} a_i b_i/d_i \text{ and } \sum_{i\in I^m} b_i^2/d_i$$

are updated during the corresponding steps 14, 16 and 18 then $z(t^L)$ and $z(t^U)$ can be calculated in $O(|I|)$ steps.

References

[1] A. Aho, J. Hopcroft and J. Ullman, *The design and analysis of computer algorithms*, Addison-Wesley, Reading, MA, 1974.

[2] R. Helgason, J. Kennington and H. Lall, "A polynomially bounded algorithm for a single constrainted quadratic program", *Mathematical Programming* **18**, 338–343 (1980).